Ever increasing software subsystem complexity and the arrival of intelligent software is producing a change in the domain to which V&V efforts are being applied that is straining traditional V&V approaches and will eventually break them. Without significant automation, traditional V&V approaches can not scale to the changing domain. To prepare for and navigate this change, NASA IV&V has chosen to embrace dynamic testing of software subsystems as a primary methodology and to make a significant investment in increasing the facility's dynamic testing capabilities via the Reconfigurable Environment for Analysis and Test of Software Systems (REATSS) project. The goals of the REATSS project are to develop an advanced dynamic simulation environment that is tailored to V&V functional, process and fiscal needs, to deliver and maintain that environment, to provide targeted configurations of that environment to specific projects as needed, to provide training in its use, and, in general, to ensure the facility has the technical capabilities to V&V future software subsystems.

The REATSS system will be technically scalable to meet any demand. Allocation and prioritization is planned to be performed by a NASA led CCB board with representation from the REATSS contractor and IV&V contractors.

Prior to initial capability, assistance will be provided to select IV&V projects to utilize off the shelf technologies to perform dynamic simulation pilots. The projects that will participate in these pilot efforts designed to start building the general dynamic simulation knowledge base will be determined by the REATSS CCB.

Initial production capability is expected to be deployed in 2006 when approved by the REATSS CCB. After initial deployment, additional capabilities will be incrementally added when approved by the REATSS CCB.

The REATSS tool suite will support an automatically distributed simulation environment supporting reusable object-oriented components allowing it to be rapidly reconfigured to execute new target subsystems within a virtual equivalent of their planned environment for the purpose of V&V of those subsystems. The subsystems may be models, simulations incorporating multiple models and simulations, actual software executing on virtual hardware via software emulation, or actual software executing on equivalent or actual target hardware. The virtual environment will consist of a mix of simulations and models of varying fidelity, emulations, equivalent hardware, and target hardware as appropriate for the subsystem under test and the goals of the test. V&V oriented process and test support will be deeply integrated into the environment including

- support for interactive test with live virtual instrumentation and traditional software debugging, tracing and profiling capabilities,
- support for batch testing using auto or manually generated tests with triggered results collection,
- a deeply integrated versioned central repository that will both support and track all aspects of REATSS usage including simulation configuration, test configuration, and results, and
- integrated IV&V process support.

The REATSS contractor will provide and maintain a scalable server farm that will initially provide necessary central system services including control and repository functions and compute engines per the expected initial simulation load. The scaling of the server farm will be scheduled by the REATSS CCB. A REATSS simulation is distributed at run time between the IV&V analysts' desktop computers and the server farm. The automatic distribution will monitor both the server farm and the participating analysts' desktop computer capabilities and load and attempt to determine the distribution that will provide the best balance between user's needs. The minimum desktop configuration is dependent on the analysts' dynamic visualization needs. Dynamic visualizations are REATSS components that run directly on the analyst's desktops. Batch mode and simple dynamic interactive execution will be possible with very minimal machines.

An advanced interactive workstation may also be provided and made available as directed and prioritized by the REATSS CCB. The facility containing shared interactive equipment is expected to be developed and expanded over time to include not only advanced interactive workstations, but also specialized I/O hardware such as cockpit mockups and interfaces to flight hardware under test or otherwise participating in a test scenario.

There are several probable groups of REATSS developers/users. The groups are created by limitations in expensive resources and/or by the level of training the user has received. Some of these are as follows:

Capability	REATSS Contractor	Component Dev	Test Dev	Sim Dev	Base User
Execute simulations, make minor simulation modifications, create tests via REATSS native tools	X	X	X	X	X
Make major modifications to simulations, including creating new simulations that use existing components	X	X	X	X	
Create tests using specialized off the shelf test generation tools	X		X		
Develop low to medium complexity components to be employed in simulations	X	X			
Develop complex components to be employed in simulations	X				
Develop framework modifications and extensions	X				

It is expected that the bulk of the users will be base users. The REATSS prime contractor is working hard to make as much capability as possible available to the base user class. The next largest group will be simulation developers. These differ from the base users in level of training and/or experience with REATSS only. Test developers are users who have been allocated and trained in the usage of a per seat licensed tool such as T-VEC and REACTIS for the creation of advanced tests. Component developers are users who have been trained in the maintenance and development of simple to medium scale REATSS components. Components must be developed and tested to a high standard of quality prior to deployment on the REATSS server farm. An incorrectly developed component could cause instability for all users as well as endanger IV&V results. Also, many tools which may be utilized by a component developer have per seat costs. A limited number of developers will be trained to this level and provided access to the tools necessary to develop components. It is expected that many component developers may also be test developers, but not all. The REATSS CCB will be charged with allocating REATSS project resources to train and equip test and component developers. Creation of major or complex components will likely be handled by the REATSS contractor. Modifications to the REATSS core architecture will be handled by the REATSS contractor.

When the REATSS CCB has determined that REATSS will be utilized on a project, decisions will be made as to how it will be utilized and the REATSS contractor will be asked to provide REATSS configuration to the IV&V project. In order to do that, the IV&V contractor will be responsible to procure and provide to the REATSS contractor any documentation, components, existing simulations, etc. specific to the subsystem, the system employing the subsystem, and the mission that might be necessary for the REATSS contractor to configure REATSS.

A full REATSS configuration might provide a complete emulation of the target processor(s) for the software subsystem, complete emulation of other processor(s) within the vehicle, full environment simulation, and ground system emulation or simulation. Such a system requires the most detailed project data, but may cost the least to develop, especially when the target processors have already been developed by a previous project such as may happen with some of the new common satellite platforms. In that case, the full configuration task may involve no more than putting new software loads on the emulations and providing for a new mission simulation. In the worst case, the documentation and products necessary to provide this REATSS configuration would be much greater.

A REATSS configuration for requirements simulation would use more simulation as opposed to emulation, require translation of the requirements into a simulation, would have lower fidelity, and would require active participation by a component developer during its use. But, it would require less project documentation to create and some of it might be reused by other REATSS configurations, including those for later phases of the project in question.

A partial list of documentation and materials that may be required by the REATSS contractor includes

- Any existing simulations or models from NASA or NASA contractors that might be useful in simulating or as an example of simulation of the subsystem itself or any system, subsystem, or environment that the subsystem under test interacts with

- A complete mission description
- Requirements documents for the subsystem under test
- Interface documents for the subsystem under test
- Source code for the subsystem under test
- Special compilers used by the subsystem developer
- Other, possibly 3rd party, software products and their requirements utilized by the subsystem such as other CSCIs not under test, operating systems, board support packages, etc.
- Binary code for the subsystem under test
- Schematics for the complete computer that runs the subsystem under test
- All or some of the above for any system that the subsystem under test interacts with including ground systems
- Any prototype, flight, or other hardware desired to be placed in the loop